

# VS-RC003 SDK Manual

Vstone Inc.

## 1. Summary

"VS-RC003 SDK" (the product) is a library for software development on the compact CPU boards "VS-RC003" and "VS-RC003HV". The CPU boards can be used with the programs RobovieMaker2 or RobovieMaker for VS-RC003 or you can use the SDK to develop software that enables the robot to load and play motion files.

Please read this manual carefully before using the product.

※ Robovie RobovieMaker is a registered trademark of Advanced Telecommunications Research Institute International.

## 2. Operation environment

### System Requirements

OS: Windows2000/XP

CPU: Pentium3 or better

RAM: 128MB or more

Interface: USB1.1 or higher

Development Environment: C language (Microsoft Visual C / C++)

Additionally, make sure the CPU board's firmware is version "1.000 revision 4" or higher. If your CPU board firmware version is older than this, use the following URL to download the latest firmware from RobovieMaker.

- VS-RC003/HV Support Page

[http://www.vstone.co.jp/top/products/robot/support\\_vsrc003.html](http://www.vstone.co.jp/top/products/robot/support_vsrc003.html)

### 2-1. Installation

To install the product on a PC, use the CD-ROM that is included. "VSRC003SDK\_Install\_\*\*\*.exe" (\*\*\*) is a placeholder for three-digit number), is the installation file. When the installer starts, set the configuration program installation folder and confirm the Agreement, to start the installation.

### 2-2. Usage

When you install the product, the development needed to run the program installation folder four files are copied "VSRC003\_SDK.h" "VSRC003\_SDK.dll" "VSRC003\_SDK.lib" "rclib.dll" (The default installation path is "C:\Program Files\VSRC003SDK\")

If including the header file "VSRC003\_SDK.h" in a project under the project menu go to (your project name) properties and in the window that pops up go to configuration properties → linker → input and in the additional dependencies box add "VSRC003\_SDK.lib". Also, if you run a program

incorporating VSRC003.h, be sure to add the files "rclib.dll" and "VSRC003\_SDK.dll" to the project folder containing the .sln file.

For information about using the sample program please see "5.Description of Sample Program ".

## 2-3.Settings for the CPU board

This product's features include the ability to configure connection settings, correct the position of the servo motors, including falls. These should be pre-configured in RobovieMaker; please write to the CPU board ROM the configuration.

For audio files, use RobovieMaker to compare files that are registered in the audio file list and the current path of the audio files recorded in the motion, this will make sure they always play the correct sound, and this product uses the value as recorded in the audio output pose sliders. To prevent drift in the output audio file, please make sure that the correct audio file list is written before you run the program.

## 3. Function Description

Describes the functions contained in the product. Each function's return value is set to follow the ERRTYPE list. For more information on the ERRTYPE list see section "4-1. ERRTYPE list description"

### **int VSRC003\_LoadMotion (TCHAR \* filename);**

RobovieMaker reads from a file created by the motion data, and writes to the CPU board's RAM. TCHAR \* filename is a pointer to a string buffer that records the file name to load. Also, write the file name including the path.

The function is successful if the return value is zero if it fails, depending on the cause it returns ERRTYPE for each member of the list. As a precaution performing this function during a motion can not be done. Use VSRC003\_GetStatus () to check whether the robot is not currently playing a motion.

### **int VSRC003\_PlayMotion (int loopnumber);**

VSRC003\_LoadMotion () is a function to play the motion currently loaded on the CPU board in a given number of loops. Use the int loopnumber to assign the number of loops. Also, if you want to set an infinite loop, use the "ENDLESS" macro as the argument.

The return value is 0 if the function is successful, if another motion is currently playing ERR\_PLAYINGMOTION is returned, if there is a problem communicating with the CPU board ERR\_COMMUNICATE is returned. As a precaution performing this function during the motion can not be done. Use VSRC003\_GetStatus () to check whether the robot is not currently playing a motion.

### **int VSRC003\_StopMotion ();**

When VSRC003\_PlayMotion () function is playing a motion, VSRC003\_StopMotion () will stop playing the motion at the next break sequence. VSRC003\_PlayMotion () loops infinitely when you set a large number of loops for the function, And if you want the motion to terminate, execute this function. If you want to force the motion to terminate execute VSRC003\_CancelMotion ().

The return value is 0 if the function is successful, if there is a problem communicating with the CPU board ERR\_COMMUNICATE is returned.

### **int VSRC003\_CancelMotion ();**

When VSRC003\_PlayMotion () function is playing a motion this function forces it to end. If you play a motion and it loops infinitely and you want to stop playing the motion immediately, execute this function. Unlike VSRC003\_StopMotion () it will not wait for next "break". It will stop motion immediately.

The return value is 0 if the function is successful, if there is a problem communicating with the CPU board ERR\_COMMUNICATE is returned.

### **int VSRC003\_GetStatus ();**

The function retrieves whether the CPU board is currently playing a motion. Before calling a function that might fail to execute if a motion is playing such as VSRC003\_LoadMotion (), or VSRC003\_PlayMotion (), this function checks whether the CPU board is playing the motion.

If the CPU board is not playing a motion the return value is 0, if a motion is currently playing ERR\_PLAYINGMOTION is returned and, if there is a problem communicating with the CPU board ERR\_COMMUNICATE is returned.

Caution, when determining if CPU board has terminated playing of the motion is "to begin the transition of the last pose, poses don't wait for the moment interpolation." Therefore, the robot may be running even when the function returns 0, if this is the case VSRC003\_LoadMotion (), VSRC003\_PlayMotion (), will not work.

### **short VSRC003\_GetValue (int valuenumber);**

The function gets the value of the variable that is sent to CPU board. The argument Int valuenumber ranges from 0 ~ 255 corresponding to a variable on the CPU board; the function returns the value of the variable in question. For variable numbers and their roles consult "VS-RC003 variable table. Pdf" (for example, if you run the function with the variable 239 the function would return the voltage of the battery.)

Note that this function does not return an error code, even if there is a communication error. Errors will be due to the function returning a number not in any real variable. So, before you perform this function if possible execute VSRC003\_GetStatus () to check whether you can communicate properly with the CPU board. In addition, this function's return type is a short. Depending on the contents of a variable of type int or unsigned short type, type casting may be necessary.

### **int VSRC003\_SetValue (int valuenumber, short value);**

VSRC003\_SetValue () is a function that assigns a value to any variable on CPU board. The argument Int valuenumber 0 ~ 255 corresponds to the variable that the argument value assigns to the variable. For variable numbers and their roles consult "VS-RC003 variable table. Pdf"

The return value is 0 if the function is successful, if there is a problem communicating with the CPU board ERR\_COMMUNICATE is returned and, if you assign int valuenumber an invalid number ERR\_IMPROPER\_MEMORY\_ACCESS is returned.

If you want to manipulate the value of a variable, an understanding of the function of that variable you change frequently is needed to avoid a failure of the robot.

### **int VSRC003\_ServoPower (int poweron\_off);**

This function turns the CPU board servo motors ON / OFF. Set int poweron\_off to 0 to turn the servo motors OFF, and set int poweron\_off to 1 to turn the servo motors ON.

The return value is 0 if the function is successful, if there is a problem communicating with the CPU board ERR\_COMMUNICATE is returned.

Note that when this function turns the servo motors on they will move to the pose last run on the CPU board. **After you start the CPU board if you are unsure of the pose last executed on the CPU board, the motors will run any motion that was on the CPU board, this can cause motor lock be safe when turning the servo motors on with the CPU board.**

## 4 Macro Description

This software has a defined list constant or macro used as an argument or function return value. And macros details of the list set are in this product are listed below.

### 4-1. Description of ERRTYPE

ERRTYPE list is set to return in case of error function. If the function returns a nonzero value, please identify the cause of the error and with the following detailed descriptions.

**ERR\_MOTION\_NOTFOUND** ----- VSRC003\_LoadMotion () file not found or the motion file given to the function failed to open. Please check the motion file given exists and the file is not corrupted.

**ERR\_MOTION\_UNKNOWNFORMAT** ----- VSRC003\_LoadMotion () indicates that the motion file could not be read. Please check RobovieMaker can read the motion file given.

**ERR\_COMMUNICATE** ----- Indicates that the function failed to communicate with the CPU board. The PC and CPU board may not be connected properly. Another reason for this error could be the board is communicating with other programs.

**ERR\_PLAYINGMOTION** ----- Indicates that the CPU board is currently playing a motion. While playing a motion VSRC003\_LoadMotion (), VSRC003\_PlayMotion (), VSRC003\_GetStatus () will return this value. In addition, VSRC003\_LoadMotion (), VSRC003\_PlayMotion () will fail when run while a motion is playing.

**ERR\_IMPROPER\_MEMORY\_ACCESS** ----- When the memory read or writes operation is performed on CPU board, indicating that access is made outside the scope of the corresponding memory address. This error occurs if the number of variables is outside the range of VSRC003\_SetValue () function.

### 4-2. Macro description

This product has been set the constants defined in the macro as an argument to a special function. Macros defined in this product are listed below.

**ENDLESS** ----- When VSRC003\_PlayMotion () is given this macro as an argument, the number of loops is set to "-1" which is interpreted as an infinite loop when the motion is set to play.

## 5. Sample Program Description

This product has two kinds of sample programs are included. This section describes their respective uses.

## 5-1. About the Sample program "VSRC003\_SDK\_SAMPLE"

"VSRC003\_SDK\_SAMPLE" reads the motion to use the product - and plays the servo motor ON / OFF function with minimal sample programs. Development and execution files of the program are in the folder where you installed the product "VSRC003\_SDK\_SAMPLE."

When you run the program it performs a specified starting motion of the robot pose to initialize first. Here, the select the file "upright" to open and it will start the appropriate motion.

When you start playing the motion, and then the robot plays motion which the file specifies. So just open the same file selection dialog, please select the motion file you want to play and open it.

Finally, you will be asked to enter a loop count of the motion when you play it, then please enter the number of times you want the motion to play from the keyboard and press Enter.

If successful the servo motor is turned ON to start playing the motion.

Please refer to the sample source for a detailed description of the program.

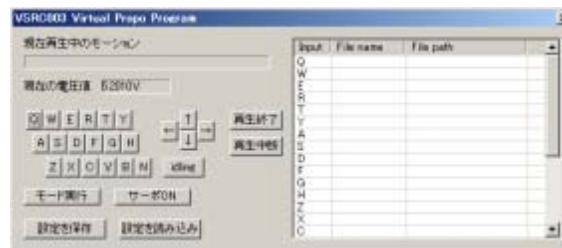
## 5-2. About the sample program "VSRC003\_VPROPO"

"VSRC003\_VPROPO" resembles the PC robot controller. It is a sample program to reproduce the robot motion in response to keyboard input. Development and execution files of the program are in the folder where you installed the product "VSRC003\_VPROPO."

When you run the program, first select the start motion to reproduce the to the CPU board. Which opens the following dialog box click "OK", and then from the file selection dialog that appears, followed by "vertical" Please select a motion to open and start.



If successful the following dialog appears, loading of motion to play starts. Steering mode selection, the starting motion used to steer the servo motors ON / OFF are done from this dialog. The following describes the details of how to operate.



### Motion Assignment

"Q" "W" "E" ... "V" "B" "N", "↑" "↓" "←" "→", "idling" when you click each button, each key on the keyboard opens a dialog box to select a motion to assign. Assign each button on the keyboard a motion here, later in pilot mode; you can maneuver the robot from the keyboard with the assigned motions. The "idling" motion is set to play when nothing is pressed.

The list on the right will be display a list of which key is assigned to each motion.

### Servo motor ON / OFF

"Servo ON" button and the servo motor ON / OFF switch. If the state where the button is pushed, the servo motor that indicates ON, when the servo motors are ON make a note not to motor lock.

### **Start maneuver**

When the "Run" mode button is clicked and the "pilot mode" ON / OFF switch steering mode that indicates ON. The steering mode is then ON, the motion assigned to each key can be played by pressing a key on the keyboard. If the motion has a loop structure, it will continue to loop through the motions as long as the key is held. The "Escape" key will open the dialog and the "end of playback" will quit playing the motion, the "Delete" key will open the dialog and the "interruption" will kill a motion that is currently playing. "Currently playing motion" will be displayed under the motion that is currently playing.

### **Load Save Settings**

The "Save settings" button will save settings to a settings motion file with the current motions assigned to each key. The "Load Settings" button will load the saved configuration file.

Also, there's a dialogue box called "current voltage values" when ON and in pilot mode it is a VSRC003\_GetValue () function which obtains voltage current from the CPU board, and converts mV units to V and displays the converted units.

※Hardware specifications, if the input voltage is 10V or higher the voltage does not display properly.

## **6. Contact**

If you experience any problems while using this product, Please check the causes and solutions are listed below. If the problem isn't listed, please contact us.

### **Q: Can not communicate properly with the CPU board.**

**A1:** CPU board may have to communicate with other programs such as RobovieMaker. Exit all other programs that could communicate with the CPU board, and try to run the program again.

**A2:** You might be running the demo CPU board and auto-mapping operation. Please try to run the program once and then reset the CPU board.

### **Q: Able to communicate with the CPU board, but the motion fails to play/load.**

**A1:** You might be running the demo CPU board and auto-mapping operation. Please try to run the program once and then reset the CPU board.

### **Q: Connection is suddenly interrupted or switched during communication with the CPU board, another motion plays.**

**A1:** You might be running the demo CPU board and auto-mapping operation. Please try to run the program once and then reset the CPU board.

### **Q: The servo motor is ON servo motor but does not move.**

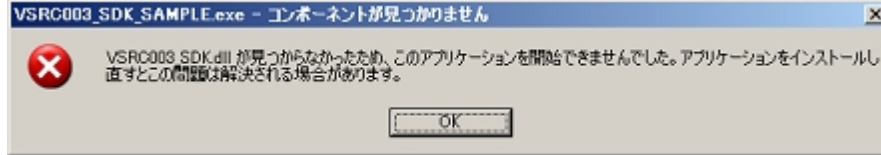
**A1:** Please make sure that power is supplied from an external source such as a battery to the CPU board.

**A2:** You might be running the demo CPU board and auto-mapping operation. Please try to run the program once and then reset the CPU board.

### **Q: The audio is actually playing a different audio file than the one saved in the motion.**

**A1:** this manual, "2-3.Settings for the CPU board" as described in the motion to play in this product, RobovieMaker will do any number of speeches correctly. When you play the correct sound, Please write a list of audio data from the motion with the same settings when you save in RobovieMaker.

**Q: The program does not start the following dialog box appears when you run the program.**



**A1:** When you run the program necessary to incorporate this product it might not find the DLL. In the same folder as the executable file place the files "rclib.dll" and "VSRC003\_SDK.dll" and then try to run the program again.

**Q: RobovieMaker2 does not play the motion you created correctly.**

**A1:** This CPU board firmware "Version 4 Revision 1" might be an older version. As described in this manual, "2.Requirements ", download the latest firmware from the support page, and update the firmware on the CPU board.

Contact Us

Vstone Inc.

Shimaya 〒 554-0024 4-4-11 Konohana-ku, Osaka

Tel: 06-6467-6601 Fax: 06-6467-6602

e-mail: [infodesk@vstone.co.jp](mailto:infodesk@vstone.co.jp) URL: <http://www.vstone.co.jp/>

(2007.11.16)